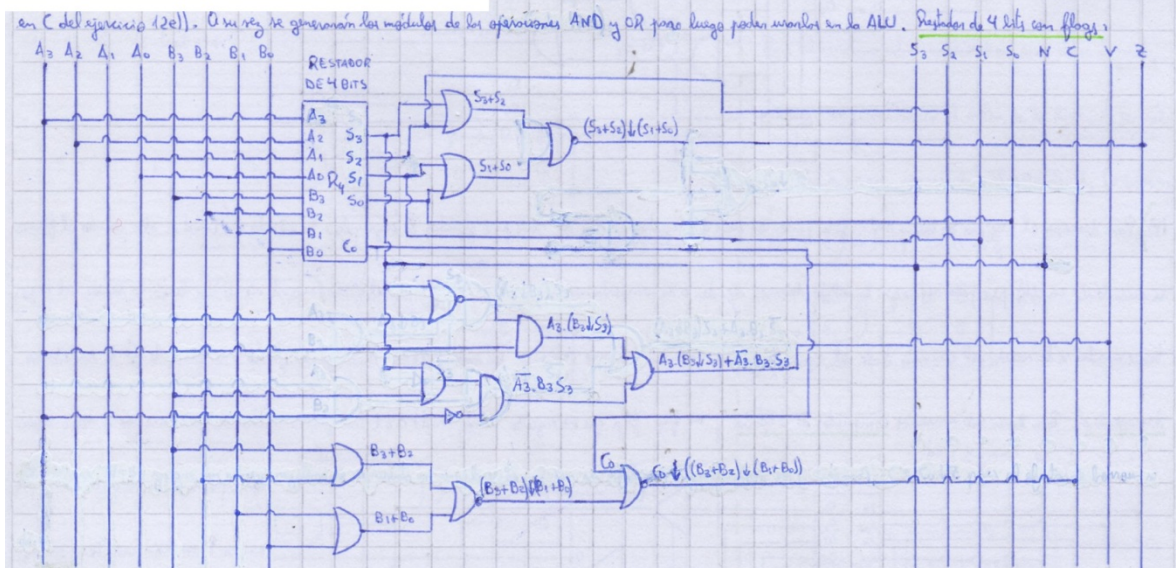


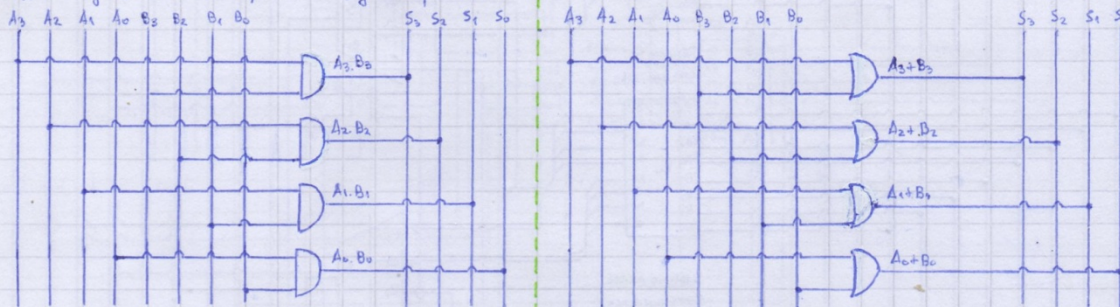
Ejercicio 20:

20) Para realizar el ALU tomaremos el módulo del circuito sumador de 4 bits con sus flags de C, V, N y Z respectivamente, proveniente del ejercicio 12b), y el módulo restador de 4 bits en complemento a 2 del ejercicio 12d) (se hará nuevamente el circuito con los flags correspondientes a complemento a 2 por lo explicado en la anotación, lo cual incluye el flag de borrow en C del ejercicio 12e)).



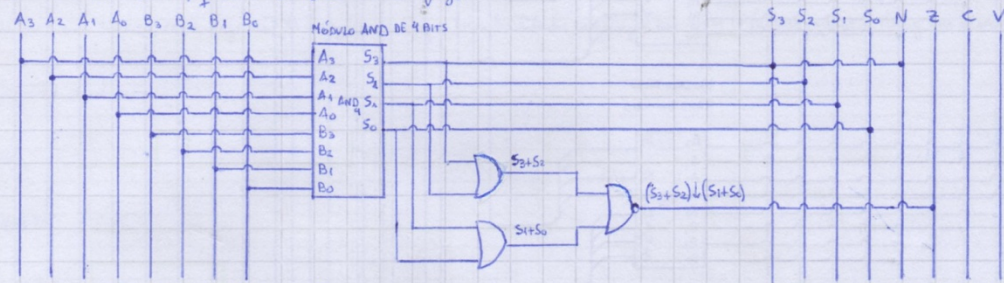
Notar que $N = S_3$, ya que un número negativo se representa en complemento a 2 cuando su cifra más significativa es 1. Además, $C = \overline{C_0} \cdot \overline{B_3} \cdot \overline{B_2} \cdot \overline{B_1} \cdot \overline{B_0}$ (al efecto borrow se produce cuando en la resta A-B no hay bit de acarreo y a su vez B ≠ 0, lo cual ya fue comentado) = $C_0 \cdot \overline{B_3} \cdot \overline{B_2} \cdot \overline{B_1} \cdot \overline{B_0}$ (ley de Morgan y definición de NOR) = $C_0 \cdot \overline{(B_3 + B_2 + B_1 + B_0)}$ (ley de Morgan) = $C_0 \cdot \overline{(B_3 + B_2) + (B_1 + B_0)}$ (ley de Morgan y definición de NOR). El overflow (V) se produce si se resta un positivo con un negativo y el resultado es negativo ($A_3 \cdot B_3 \cdot S_3 = A_3 \cdot B_3 \cdot S_3$ (ley de Morgan y definición de NOR)) y viceversa ($\overline{A_3} \cdot \overline{B_3} \cdot S_3$), quedando $V = A_3 \cdot B_3 \cdot S_3 + \overline{A_3} \cdot \overline{B_3} \cdot S_3$. El flag Z da 1 si el resultado es 0: $Z = \overline{S_0} \cdot \overline{S_1} \cdot \overline{S_2} \cdot \overline{S_3}$ (ley de Morgan) = $\overline{(S_0 + S_1) + (S_2 + S_3)}$ (ley de Morgan) = $\overline{(S_0 + S_1)} \cdot \overline{(S_2 + S_3)}$ (definición de NOR).

Otro que se genera el módulo de la operación AND, y el de la operación OR:

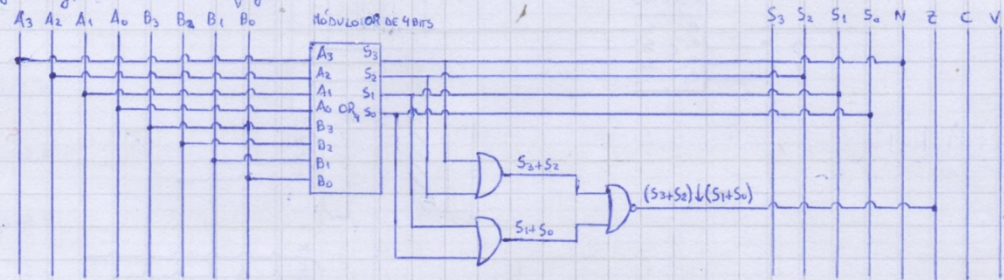


Entonces, para dicho módulo genero una versión con los flags N, C, Z, V, donde sea que como todos los resultados son representables y no hay bit de acarreo, luego

mis valores son cero. O sea, queda el módulo de AND con flags:



Y a su vez, el módulo de OR con flags:



Juntamos los 4 módulos de operaciones para formar el circuito interno de la ALU, donde tengo los entornos A3...A0, B3...B0 para representar A y B, y a su vez tengo una

entrada correspondiente a cada operación: add (suma), sub (resta), and, or. Haciéndolo uno de los entornos podemos decidir qué operación realizar, pero al tener más de

una en simultáneo se pueda generar un conflicto, por lo que restringiré ese caso para que si ocurre al querer guardarlos en el registro, la salida no se sea afectada.

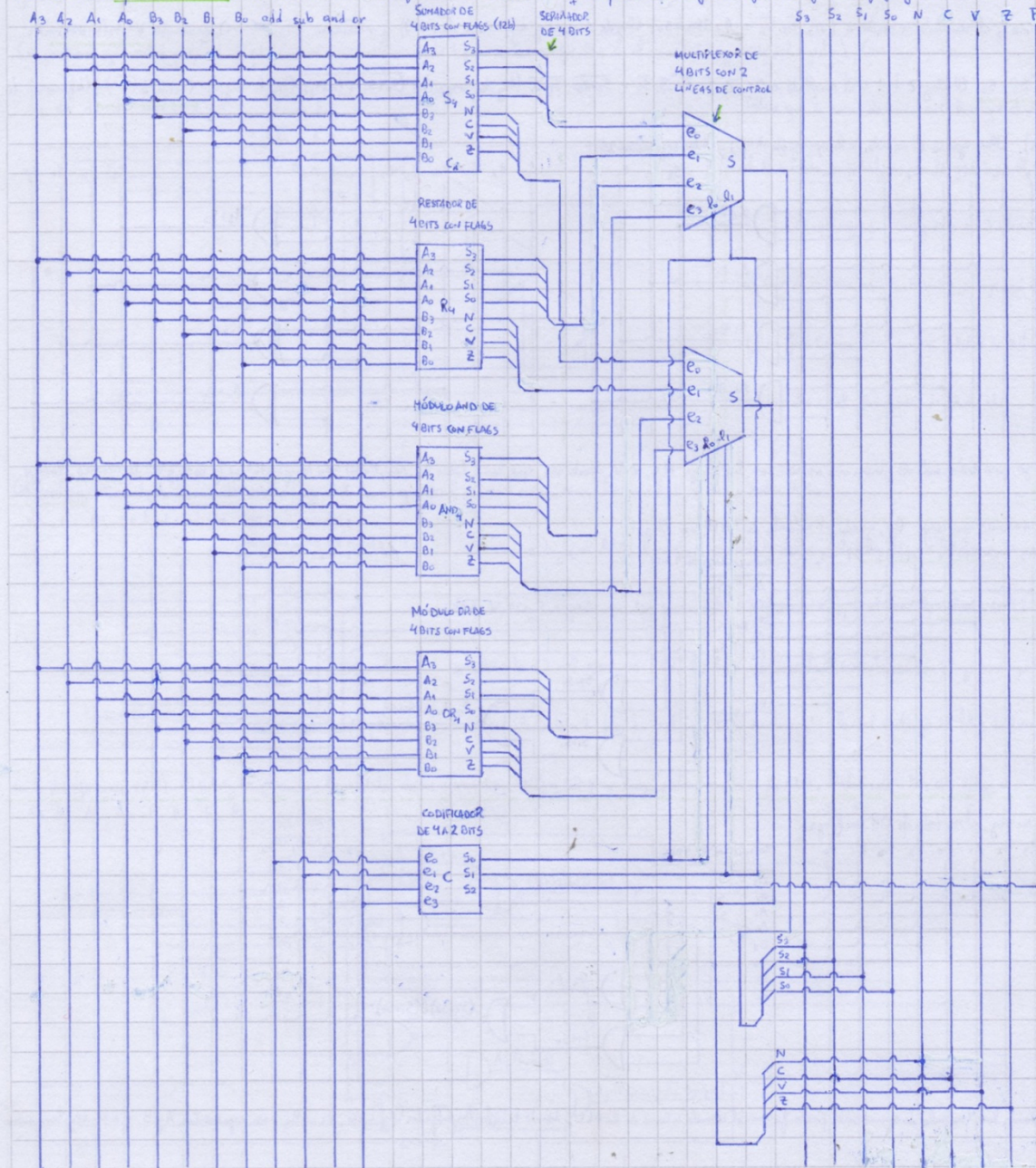
Otro, como en cada operación tenemos los valores S3...S1, N, C, V, Z y según la entrada seleccionada tendremos uno u otro, usaremos un codificador para recibir cada entor-

do con un minimal de 2 bits (add → 00, sub → 01, and → 10, or → 11), como el implementado en B0, pues la salida extra S2 nos permitirá determinar si dos entornos

de operación están activos en simultáneo o si ninguno lo está, para luego usarlo como notificación a guardar en el registro de salida. Los minimales valores del codificador serán

Las líneas de control de 2 multiplexores de codones de 4 bits, donde en uno elegiremos entre los codones S_3, S_2, S_1, S_0 , y en el otro N, C, V, Z , que puede formar y reparar

a través de los registradores de 4 bits. La salida S_2 del codificador sirve como una salida más que me permitirá restringir la carga a los registros de flaps y salida del circuito macro.



Otro, con este módulo del circuito interno de la ALU puedo armar el circuito macro con dos registros simples para guardar los valores de las entradas A y B, los 4
entradas correspondientes a las operaciones, un registro de salida restringido para la salida del ALU y 4 salidas para los flaps salientes del ALU. Los registros
simples requieren de una entrada Load adicional para cargar el valor en cada uno (una para cada uno), mientras que en el de salida restringida la entrada Load está
no determinado por la salida F negativa (carga el valor en el registro si se selecciona una sola entrada de operación), pero se necesita una entrada Load para los su re

